



Benchmarking the BFGS Algorithm on the BBOB-2009 Noisy Testbed

Raymond Ros

► To cite this version:

Raymond Ros. Benchmarking the BFGS Algorithm on the BBOB-2009 Noisy Testbed. GECCO, Jul 2009, Montréal, Canada. inria-00377079v2

HAL Id: inria-00377079

<https://inria.hal.science/inria-00377079v2>

Submitted on 22 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Benchmarking the BFGS Algorithm on the BBOB-2009 Noisy Testbed

Raymond Ros
Univ. Paris-Sud, LRI
UMR 8623 / INRIA Saclay, projet TAO
F-91405 Orsay, France
raymond.ros@lri.fr

ABSTRACT

The BFGS quasi-Newton method is benchmarked on the noisy BBOB-2009 testbed. A multistart strategy is applied with a maximum number of function evaluations of about 10^4 times the search space dimension.

Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization, BFGS, Quasi-newton

1. INTRODUCTION

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [1, 3, 4, 7] is a real-parameter unconstrained non-linear optimization method. The BFGS method belongs to the class of quasi-Newton methods which, by supposing that the objective function can be locally approximated by a quadratic function near its optimum, tries to approximate the Hessian matrix of this quadratic function. The BFGS method is tested here in its default setting on a testbed of noisy functions.

2. ALGORITHM PRESENTATION

Quasi-Newton methods address the problem of unconstrained black-box optimization by the determination of the stationary point of a function using a second-order approximation. The Hessian matrix is iteratively approximated by finding a search direction using the gradient of the current point and operating a line search to find the step size. For

each iteration $D + 1$ function evaluations (where D denotes the problem dimension) or double if the step size changes, are done to evaluate the gradient and the function value at the current point. A simple stochastic independent restart procedure (as advised in [5]) was added.

3. EXPERIMENTAL PROCEDURE

The Matlab implementation of the BFGS method was used. It is accessible using the generic function `fminunc` (revision 1.1.6.3) that proposes, among others, the BFGS method for the update of the Hessian Matrix. The starting point is chosen uniformly in $[-5, 5]^D$. The stopping criteria were chosen such that a restart occurs due to numerical errors. The multistart strategy was used with at most 100 restarts to reduce the duration of an experiment. For the same reason, a run is limited to at most $10^4 \times D$ function evaluations. The entire experiment took roughly 7 hours. The algorithm used is presented in Figure 1. No parameter tuning was done, the CrE [5] is computed to zero.

4. RESULTS AND DISCUSSION

Results from experiments according to [5] on the benchmarks functions given in [2, 6] are presented in Figures 2 and 3 and in Tables 1 and 2. The proposed algorithm only solves 5, 3, 2, 2, 2 and 1 out of 30 functions in dimension 2, 3, 5, 10, 20 and 40 respectively. The functions solved use the Cauchy noise model only. The Rosenbrock function and Gallagher functions are solved in dimension 2. The sphere with the Cauchy noise is solved for all dimensions tested, though the algorithm scales comparatively worse than on the noiseless sphere. A more severe Cauchy noise model decreases the probability of success of the algorithm. No success was obtained on the multimodal functions with severe noise.

5. CPU TIMING EXPERIMENT

For the timing experiment, the proposed algorithm was run on f_8 and restarted until at least 30 seconds have passed (according to Figure 2 in [5]). The experiments were conducted with an Intel Core 2 6700 processor (2.66GHz) with Matlab R2008a on Linux 2.6.24.7. The results were 6.0, 4.7, 3.7, 3.0, 2.9, 2.9 and 2.8×10^{-4} seconds per function evaluations in dimension 2, 3, 5, 10, 20, 40 and 80 respectively.

6. CONCLUSION

The results of a quasi-Newton method algorithm with restarts were presented. They show the limits of a default Quasi-Newton method on noisy testbeds.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-505-5/09/07 ...\$5.00.

f101 in 5-D, N=15, mFE=18156						f101 in 20-D, N=15, mFE=65163						f102 in 5-D, N=15, mFE=18522						f102 in 20-D, N=15, mFE=58506					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	12	8.2e3	5.7e3	1.0e4	5.7e3	0	11e+1	85e+0	15e+1	2.0e4	10	12	9.9e3	8.1e3	1.3e4	8.6e3	0	12e+1	87e+0	15e+1	2.5e4		
1	0	70e-1	35e-1	11e+0	7.9e3	1	0	59e-1	28e-1	11e+0	8.9e3		
1e-1	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		
f103 in 5-D, N=15, mFE=654						f103 in 20-D, N=15, mFE=11970						f104 in 5-D, N=15, mFE=8814						f104 in 20-D, N=15, mFE=22659					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	15	8.2e1	5.0e1	1.1e2	8.2e1	15	3.0e3	2.0e3	3.8e3	3.0e3	10	0	61e+1	24e+1	22e+2	4.0e3	0	13e+4	85e+3	21e+4	1.0e4		
1	15	1.2e2	5.1e1	1.5e2	1.2e2	15	3.5e3	2.3e3	4.6e3	3.5e3	1		
1e-1	15	1.2e2	4.3e1	1.9e2	1.2e2	15	3.7e3	2.5e3	4.3e3	3.7e3	1e-1		
1e-3	15	1.2e2	5.1e1	2.2e2	1.2e2	15	3.7e3	2.5e3	4.7e3	3.7e3	1e-3		
1e-5	15	1.2e2	4.2e1	1.4e2	1.2e2	15	3.7e3	2.9e3	4.6e3	3.7e3	1e-5		
1e-8	15	1.2e2	7.5e1	1.6e2	1.2e2	15	3.7e3	2.1e3	4.2e3	3.7e3	1e-8		
f105 in 5-D, N=15, mFE=7668						f105 in 20-D, N=15, mFE=21462						f106 in 5-D, N=15, mFE=26574						f106 in 20-D, N=15, mFE=85008					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	0	82e+1	39e+1	21e+2	3.2e3	0	15e+4	10e+4	22e+4	1.0e4	10	15	1.8e3	1.5e3	2.3e3	1.8e3	0	91e+1	34e+1	23e+2	3.2e4		
1	1	9	2.9e4	2.5e4	3.3e4	1.7e4		
1e-1	1e-1	3	1.1e5	9.6e4	1.2e5	1.8e4		
1e-3	1e-3	0	67e-2	20e-3	28e-1	1.4e4		
1e-5	1e-5		
1e-8	1e-8		
f107 in 5-D, N=15, mFE=11610						f107 in 20-D, N=15, mFE=26712						f108 in 5-D, N=15, mFE=4530						f108 in 20-D, N=15, mFE=16443					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	12	5.9e3	5.2e3	7.2e3	3.4e3	0	11e+1	89e+0	14e+1	1.3e4	10	10	4.9e3	4.4e3	5.4e3	2.7e3	0	11e+1	95e+0	14e+1	5.0e3		
1	0	61e-1	17e-1	10e+0	5.6e3	1	0	84e-1	44e-1	16e+0	2.8e3		
1e-1	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		
f109 in 5-D, N=15, mFE=4422						f109 in 20-D, N=15, mFE=48111						f110 in 5-D, N=15, mFE=5352						f110 in 20-D, N=15, mFE=12390					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	15	4.3e2	3.9e2	5.3e2	4.3e2	3	2.0e5	2.0e5	2.2e5	2.9e4	10	0	73e+1	39e+1	18e+2	1.4e3	0	14e+4	96e+3	22e+4	5.0e3		
1	15	7.2e2	5.3e2	1.0e3	7.2e2	2	3.1e5	2.7e5	3.4e5	2.7e4	1		
1e-1	15	7.4e2	5.7e2	9.6e2	7.4e2	1	6.4e5	6.0e5	6.8e5	6.5e3	1e-1		
1e-3	15	9.5e2	6.5e2	1.2e3	9.5e2	1	6.4e5	6.1e5	6.8e5	6.5e3	1e-3		
1e-5	15	9.5e2	5.9e2	1.2e3	9.5e2	1	6.4e5	6.0e5	6.9e5	6.5e3	1e-5		
1e-8	15	9.5e2	4.9e2	1.1e3	9.5e2	1	6.4e5	5.6e5	6.9e5	6.5e3	1e-8		
f111 in 5-D, N=15, mFE=2988						f111 in 20-D, N=15, mFE=9366						f112 in 5-D, N=15, mFE=14580						f112 in 20-D, N=15, mFE=45444					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	0	10e+2	13e+1	33e+2	1.8e3	0	15e+4	88e+3	20e+4	5.0e3	10	2	9.8e4	9.3e4	1.0e5	1.3e4	0	14e+4	30e+3	22e+4	2.0e4		
1	1	0	92e+0	68e-1	45e+1	5.6e3		
1e-1	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		
f113 in 5-D, N=15, mFE=7968						f113 in 20-D, N=15, mFE=22911						f114 in 5-D, N=15, mFE=4074						f114 in 20-D, N=15, mFE=14406					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	4	2.4e4	2.1e4	2.7e4	6.1e3	0	58e+1	46e+1	85e+1	8.9e3	10	0	28e+0	12e+0	42e+0	2.2e3	0	67e+1	39e+1	10e+2	4.5e3		
1	0	24e+0	54e-1	42e+0	2.5e3	1		
1e-1	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		
f115 in 5-D, N=15, mFE=10314						f115 in 20-D, N=15, mFE=46410						f116 in 5-D, N=15, mFE=5106						f116 in 20-D, N=15, mFE=13944					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	1	1.4e5	1.4e5	1.5e5	6.5e3	0	62e+1	42e+1	11e+2	2.5e4	10	0	12e+2	45e+1	31e+2	2.2e3	0	55e+3	40e+3	71e+3	7.9e3		
1	0	24e+0	11e+0	54e+0	5.0e3	1		
1e-1	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		
f117 in 5-D, N=15, mFE=2970						f117 in 20-D, N=15, mFE=10038						f118 in 5-D, N=15, mFE=14562						f118 in 20-D, N=15, mFE=47691					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	0	91e+1	25e+1	26e+2	1.6e3	0	49e+3	26e+3	68e+3	3.2e3	10	0	31e+1	69e+0	69e+1	8.9e3	0	46e+3	30e+3	78e+3	1.8e4		
1	1		
1e-1	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		
f119 in 5-D, N=15, mFE=13068						f119 in 20-D, N=15, mFE=28077						f120 in 5-D, N=15, mFE=4584						f120 in 20-D, N=15, mFE=16884					
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}		
10	15	2.6e3	1.6e3	3.7e3	2.6e3	0	43e+0	29e+0	50e+0	1.3e4	10	15	6.4e2	5.6e2	1.1e3	6.4e2	0	34e+0	24e+0	44e+0	1.3e4		
1	2	8.2e4	7.9e4	8.6e4	1.2e4	1	0	37e-1	14e-1	76e-1	8.9e2		
1e-1	0	50e-1	83e-2	76e-1	7.9e3	1e-1		
1e-3	1e-3		
1e-5	1e-5		
1e-8	1e-8		

Figure 1: Matlab code: Multistart BFGS

```
function [x, ilaunch] = MY_OPTIMIZER(FUN, D, ftarget, maxfunevals)
% minimizes FUN in D dimensions by independent restarts of fminunc (BFGS).
% ftarget and maxfunevals are additional external termination conditions.
% Search space is [-5, 5]^D

% set options, make sure we always terminate
options = optimset('fminunc');
options = optimset(options, 'LargeScale', 'off'); % BFGS algorithm
options = optimset(options, 'MaxIter', inf, 'Tolfun', 1e-11, 'TolX', 0, ...
    'OutputFcn', @callback, 'Display', 'off');

maxfunevals = min(1e4*DIM, maxfunevals);

% multistart such that ftarget is reached with reasonable prob.
for ilaunch = 1:100; % relaunch optimizer up to 100 times
    options = optimset(options, 'MaxFunEvals', ...
        maxfunevals - feval(FUN, 'evaluations'));
    x = fminunc(FUN, 10*rand(DIM,1)-5, options);
    if (feval(FUN, 'fbest') < ftarget || ...
        feval(FUN, 'evaluations') >= maxfunevals)
        break;
    end
end

function stop = callback(x, optimValues, state)
    stop = false;
    if optimValues.fval < ftarget
        stop = true;
    end
end
end % function
```

Acknowledgments

The first author would like to acknowledge the support, help, and work of the BBOB team with particular kudos to Anne Auger, Steffen Finck and Nikolaus Hansen.

7. REFERENCES

- [1] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.
- [2] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noisy functions. Technical Report 2009/21, Research Center PPE, 2009.
- [3] R. Fletcher. A new approach to variable metric algorithms. *Computer journal*, 13:317–322, 1970.
- [4] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.
- [5] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.
- [6] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noisy functions definitions. Technical Report RR-6869, INRIA, 2009.
- [7] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.

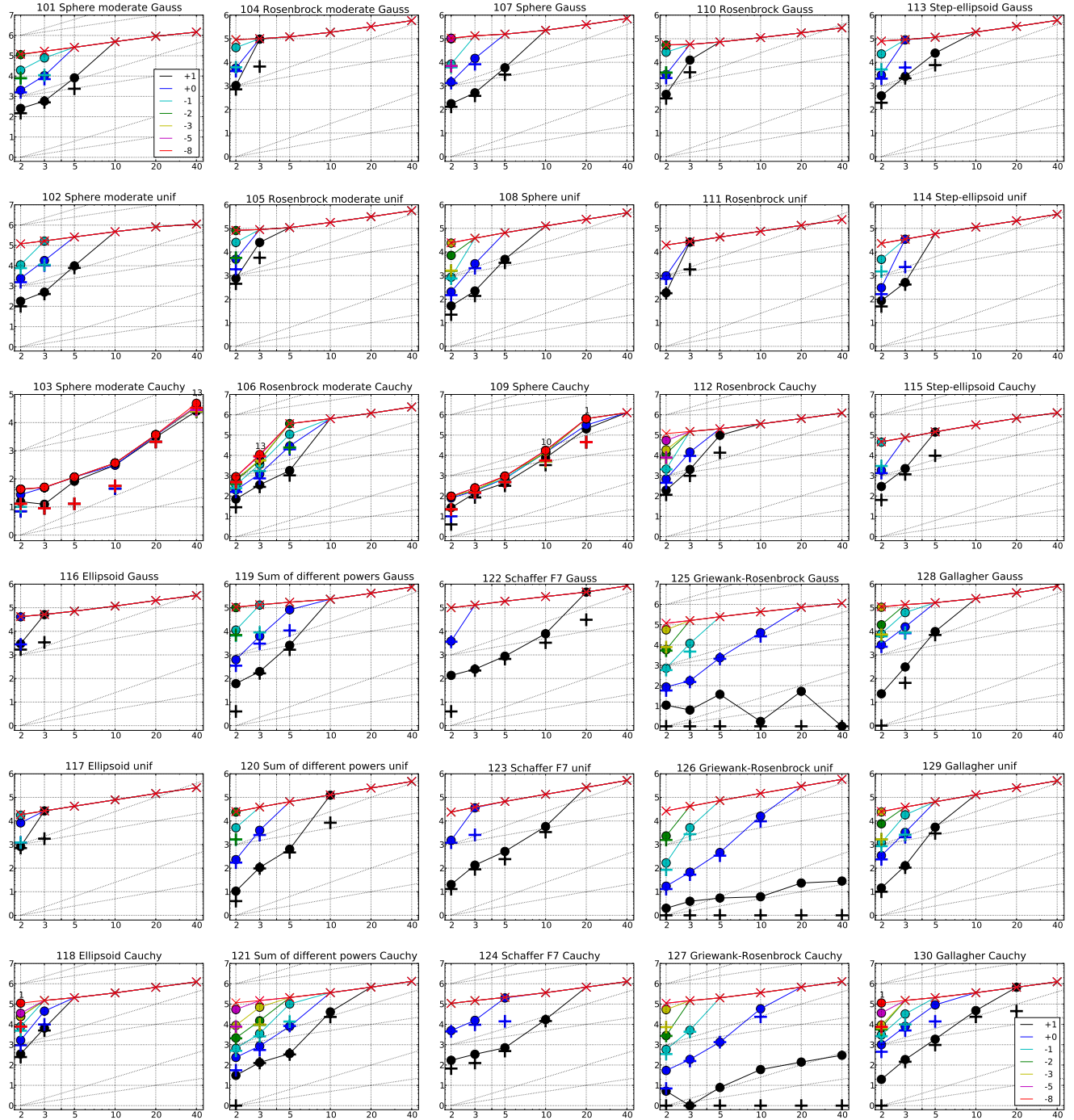


Figure 2: Expected Running Time (ERT, \bullet) to reach $f_{\text{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in log-log presentation. The $\text{ERT}(\Delta f)$ equals to $\#FES(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\text{opt}} + \Delta f$ was surpassed during the trial. The $\#FES(\Delta f)$ are the total number of function evaluations while $f_{\text{opt}} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and f_{opt} denotes the optimal function value. Crosses (\times) indicate the total number of function evaluations $\#FES(-\infty)$. Numbers above ERT-symbols indicate the number of successful trials. Annotated numbers on the ordinate are decimal logarithms. Additional grid lines show linear and quadratic scaling.

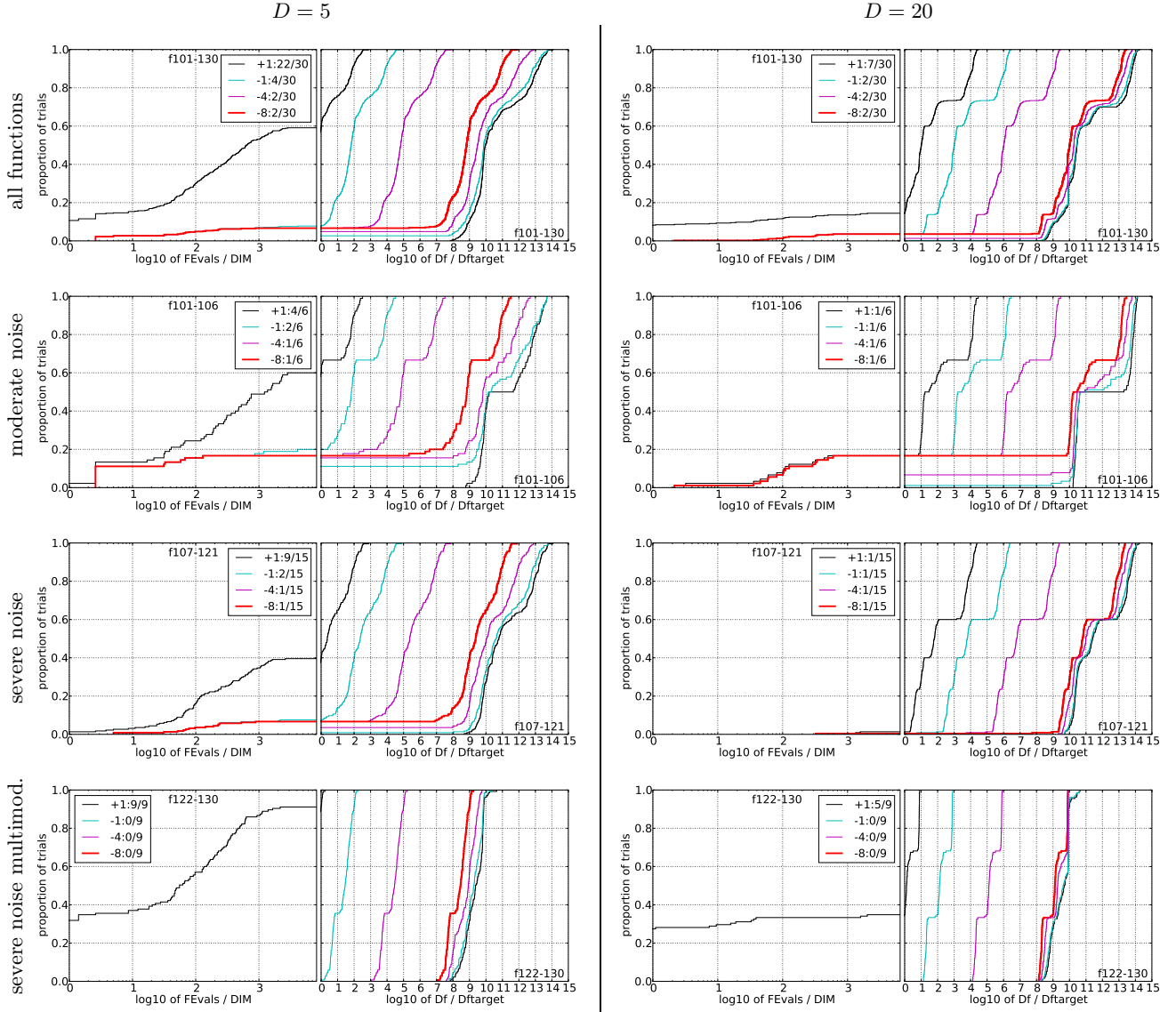


Figure 3: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D , to fall below $f_{\text{opt}} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the best achieved Δf divided by 10^k (upper left lines in continuation of the left subplot), and best achieved Δf divided by 10^{-8} for running times of $D, 10D, 100D \dots$ function evaluations (from right to left cycling black-cyan-magenta). Top row: all results from all functions; second row: moderate noise functions; third row: severe noise functions; fourth row: severe noise and highly-multimodal functions. The legends indicate the number of functions that were solved in at least one trial. FEvals denotes number of function evaluations, D and DIM denote search space dimension, and Δf and Df denote the difference to the optimal function value.

f_{121} in 5-D, N=15, mFE=14544						f_{121} in 20-D, N=15, mFE=48426						f_{122} in 5-D, N=15, mFE=14370						f_{122} in 20-D, N=15, mFE=35028								
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}
10	15	3.6e2	3.3e2	4.4e2	3.6e2	0	33e+0	23e+0	49e+0	2.0e4	10	15	8.8e2	7.1e2	1.2e3	8.8e2	1	4.6e5	4.6e5	4.8e5	2.9e4	0	13e+0	11e+0	22e+0	1.6e4
1	14	7.9e3	6.8e3	9.5e3	7.9e3	1	0	36e-1	27e-1	69e-1	4.0e3
1e-1	2	1.0e5	9.9e4	1.0e5	1.4e4	1e-1
1e-3	0	50e-2	99e-3	92e-2	7.9e3	1e-3
1e-5	1e-5
1e-8	1e-8
f_{123} in 5-D, N=15, mFE=4764						f_{123} in 20-D, N=15, mFE=18333						f_{124} in 5-D, N=15, mFE=14898						f_{124} in 20-D, N=15, mFE=48426								
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}
10	15	5.1e2	3.0e2	7.0e2	5.1e2	0	14e+0	10e+0	21e+0	7.9e3	10	15	7.1e2	5.6e2	9.8e2	7.1e2	0	13e+0	11e+0	19e+0	2.5e4
1	0	39e-1	24e-1	49e-1	2.0e3	1	1	2.0e5	2.0e5	2.1e5	1.4e4
1e-1	1e-1	0	39e-1	15e-1	61e-1	4.0e3
1e-3	1e-3
1e-5	1e-5
1e-8	1e-8
f_{125} in 5-D, N=15, mFE=17160						f_{125} in 20-D, N=15, mFE=53130						f_{126} in 5-D, N=15, mFE=4998						f_{126} in 20-D, N=15, mFE=20181								
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}
10	15	3.7e1	1.0e0	5.1e1	3.7e1	15	5.3e1	1.0e0	1.6e2	5.3e1	10	15	5.4e0	1.4e0	1.3e1	5.4e0	15	2.3e1	1.0e0	3.6e1	2.3e1
1	15	2.4e3	1.8e3	3.5e3	2.4e3	0	21e-1	16e-1	23e-1	2.2e4	1	15	4.6e2	3.0e2	6.3e2	4.6e2	0	17e-1	14e-1	21e-1	1.0e4
1e-1	0	46e-2	33e-2	54e-2	5.6e3	1e-1	0	49e-2	26e-2	63e-2	1.8e3
1e-3	1e-3
1e-5	1e-5
1e-8	1e-8
f_{127} in 5-D, N=15, mFE=14124						f_{127} in 20-D, N=15, mFE=49329						f_{128} in 5-D, N=15, mFE=11808						f_{128} in 20-D, N=15, mFE=28938								
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}
10	15	7.8e0	1.0e0	7.8e0	7.8e0	15	1.4e2	9.9e1	1.8e2	1.4e2	10	10	9.8e3	7.7e3	1.0e4	5.8e3	0	75e+0	70e+0	79e+0	1.3e4
1	15	1.3e3	9.9e2	1.8e3	1.3e3	0	20e-1	16e-1	22e-1	3.2e4	1	0	94e-1	35e-1	12e+0	4.0e3
1e-1	0	38e-2	18e-2	63e-2	6.3e3	1e-1
1e-3	1e-3
1e-5	1e-5
1e-8	1e-8
f_{129} in 5-D, N=15, mFE=4518						f_{129} in 20-D, N=15, mFE=17346						f_{130} in 5-D, N=15, mFE=14472						f_{130} in 20-D, N=15, mFE=47733								
Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	Δf	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}	#	ERT	10%	90%	RT _{succ}
10	8	5.5e3	5.0e3	6.2e3	3.0e3	0	76e+0	71e+0	80e+0	5.6e3	10	15	1.9e3	1.5e3	2.6e3	1.9e3	1	6.7e5	6.6e5	6.8e5	4.3e4
1	0	76e-1	54e-1	13e+0	2.5e3	1	2	9.1e4	7.7e4	9.9e4	7.3e3	0	75e+0	32e+0	78e+0	2.8e4
1e-1	1e-1	0	20e-1	50e-2	57e-1	3.2e3
1e-3	1e-3
1e-5	1e-5
1e-8	1e-8

Table 2: Shown are, for functions f_{121} - f_{130} and for a given target difference to the optimal function value Δf : the number of successful trials (#); the expected running time to surpass $f_{\text{opt}} + \Delta f$ (ERT, see Figure 2); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value (RT_{succ}). If $f_{\text{opt}} + \Delta f$ was never reached, figures in *italics* denote the best achieved Δf -value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 2 for the names of functions.